渐近时间上/下界

O记号（渐近上界）

- Def : 对给定函数$g(n)$, $O(g(n))$是一个函数集合

$$\boxed{\begin{aligned} O(g(n)) = \{ f(n) \mid & \exists 常数 c, n_0 > 0, such\ that \\ & 0 \le f(n) \le cg(n)\ for\ all\ n \ge n_0 \} \end{aligned}}$$

- 即在一个常数因子范围内$g(n)$是 $f(n)$的**渐近上界**
- 注意： $f(n) = \Theta(g(n))$蕴含$f(n) = O(g(n))$，因为**Θ是一个比O记号更强的概念**

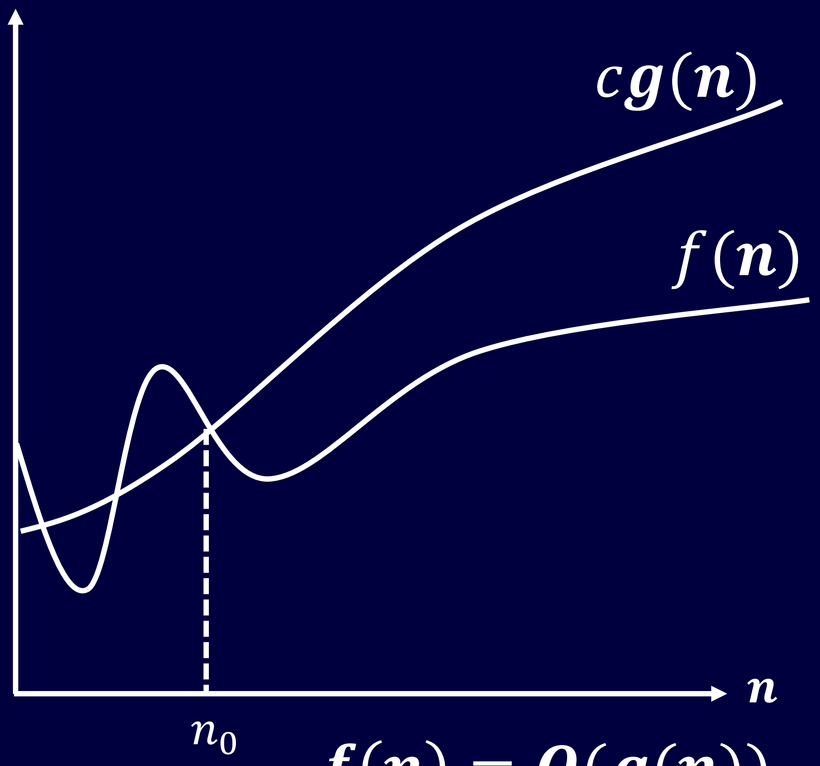
集合论角度：$\because \Theta\big(g(n)\big) \subseteq O\big(g(n)\big)$

$$\therefore f(n) = \Theta(g(n)) \Rightarrow f(n) = O(g(n))$$

O记号（渐近上界）

■ 大O的数学定义(渐近上界)

若$g(n)$和$f(n)$是定义在正整数集合上的两个函数，则$f(n)=O(g(n))$表示存在两个正的常数c和n_0，使得当$n \geq n_0$时都满足$0 \leq f(n) \leq cg(n)$。



$f(n)=O(g(n))$

- 函数$f(n)$是集合$O(g(n))$的成员;
- 我们称$g(n)$是$f(n)$的一个渐近上界(Asymptotically upper bound)，限制算法最坏情况运行时间。

- 大O记号和Θ记号的区别
  - O描述上界，当被用于界定一个最坏运行时间时，蕴含着该算法在任意输入上的运行时间都囿于此界
  - Θ则不然，一个算法的最坏运行时间是$\Theta(g(n))$，并非蕴含着该算法对每个输入实例的运行时间均囿于$\Theta(g(n))$
- 示例：插入排序
  - 在最坏情况下，数组完全逆序，算法的时间复杂度是$O(n^2)$，该界适用于算法对于每个输入的运行时间
  - 但插入排序最坏情况运行时间的界$\Theta(n^2)$并不表示算法对所有输入的运行时间的界也是$\Theta(n^2)$：当输入初始有序时，插入排序的运行时间是$\Theta(n)$，而不是$\Theta(n^2)$。

示例

$$an^3 + bn^2 + cn + d = O(n^3), a > 0$$

证明：$an^3 + bn^2 + cn + d = an^3 + bn^2 + \Theta(n)$

$$= an^3 + \Theta(n^2)$$

$$= \Theta(n^3)$$

又由于集合$\Theta(n^3)$是被$O(n^3)$包含的，故得证

示例（续）

- $\frac{1}{3}n^2 - 3n \in O(n^2)$

  证明： 令$\frac{1}{3}n^2 - 3n \leq cn^2$，则有$c \geq \frac{1}{3} - \frac{3}{\mathrm{n}}$，当$c = \frac{1}{3}$且$n > 1$ 时上式成立

- $k_1 n^2 + k_2 n + k_3 \in O(n^2)$

  证明： 令$k_1 n^2 + k_2 n + k_3 \leq (k_1 + |k_2| + |k_3|)n^2$，

  则当$c > k_1 + |k_2| + |k_3|$且$n \geq 1$时，$k_1 n^2 + k_2 n + k_3 \leq cn^2$成立

- $k_1 n^2 + k_2 n + k_3 \in O(n^3)$

  证明： $k_1 n^2 + k_2 n + k_3 \leq (k_1 + |k_2| + |k_3|)n^3$

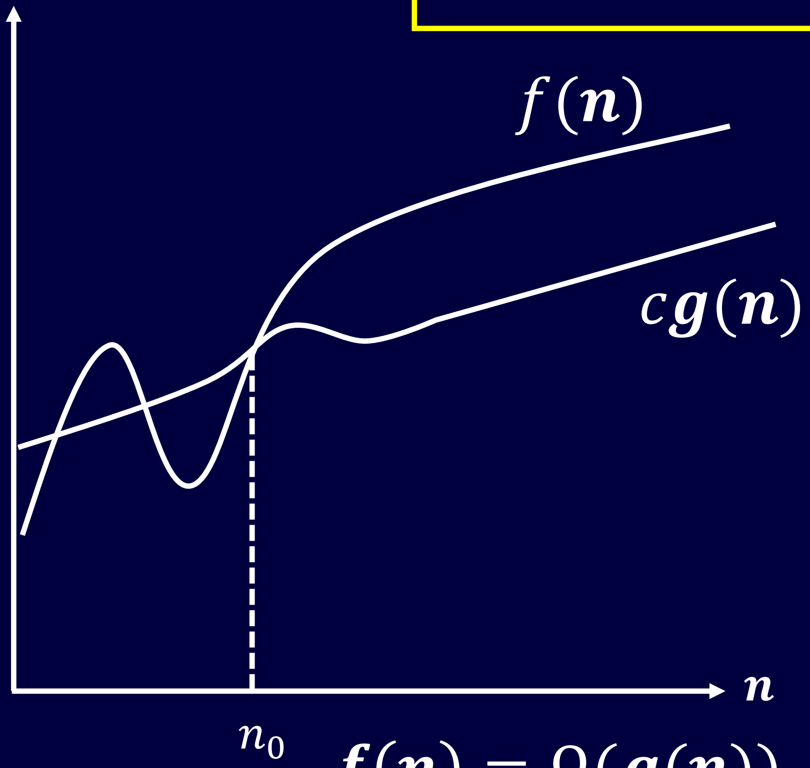
O记号使用时的注意事项

- 当说算法的运行时间上界是$O(n^2)$，往往是指其最坏运行时间，无须修饰语，对那些最好、最坏、平均时间数量级不同者均成立，而Θ则要分开表达、加修饰语。
- 使用大O表示法通常可以更容易地分析算法；比如我们可以很容易地证明插入排序的运行时间上界是$O(n^2)$。
- 一些非形式化描述：
  - 我们常用$f(n) = O(g(n))$来替代$f(n) \in O(g(n))$
  - 我们常在等式中使用$O(n)$：$2n^2 + 3n + 1 = 2n^2 + O(n)$表示$2n^2 + 3n + 1 = 2n^2 + f(n)$，其中$f(n) \in O(n)$，可消去不必要的细节，突出主项的常数因子等。
  - 常函数被写作$O(1)$

Ω记号（渐近下界）

- ***Def*** **: 对给定函数 $g(n)$, $\Omega(g(n))$ 是一个函数集合：**

$$\boxed{\begin{aligned}\Omega(g(n)) = \{ f(n) \mid\ & \exists \text{常数} c,\ n_0 > 0 \text{ such that} \\ & 0 \le cg(n) \le f(n) \text{ for all } n \ge n_0 \}\end{aligned}}$$



$f(n) = \Omega(g(n))$

- 存在正常量 c 使得对于 n_0 及其右边的所有 n 值，函数 $f(n)$ 值总大于 $cg(n)$，则 $f(n) \in \boldsymbol{\Omega(g(n))}$

- 我 们 称 $g(n)$ 是 $f(n)$ 的 **渐 近 下 界** (Asymptotically lower bound)

Ω记号（渐近下界）

- **Theorem. 1：对任意函数$f(n)$和$g(n)$，$f(n) = \Theta(g(n))$当且仅当$f(n) \in O(g(n))$和$f(n) \in \Omega(g(n))$。**

  即：$g(n)$是$f(n)$的渐紧界当且仅当$g(n)$是$f(n)$的渐近上界和渐近下界

- 当Ω用来界定一个算法的最好情况下的运行时间时，蕴含着该算法在任意输入上的运行时间都囿于此界。

- 示例：插入排序的下界是$\Omega(n)$，即对任何实例成立，插入排序的最好运行时间是$\Omega(n)$

  $$\because n = \Omega(n)\ n^2 = \Omega(n)$$

o记号（渐近非紧确上界）

- 大O记号表示的渐近上界可以是渐近紧致的，也可以是渐近非紧界

$$2n^2 = O(n^2) \because \frac{2n^2}{n^2} \to 2\text{，紧致界}$$

$$2n = O(n^2) \because \frac{2n}{n^2} \to 0\text{，非紧致界}$$

- 小o记号用来表示——函数的渐近非紧致上界

找出在形式化定义上与大O记号的两处差别

- **Def:** $$\boxed{o(g(n)) = \{ f(n) \mid \forall 常数c > 0, \exists 常数n_0 > 0 \ such\ that \ 0 \le f(n) < cg(n) \text{ for all } n \ge n_0 \}}$$

只要n足够大，$g(n)$是$f(n)$的上界。

例：$2n = o(n^2)$ 但 $2n^2 \neq o(n^2)$

直观上，当$n \to \infty$，$f(n)$相对于$g(n)$是可忽略的。

即：$f(n) = o(g(n))$蕴含着$\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = 0$

或者说f和g数量级不同，否则不可能对于任意常数c，都有$cg(n)$严格大于$f(n)$。

ω记号（渐近非紧确下界）

找出在形式化定义上与大Ω记号的两处差别

- **Def**: $\omega(g(n)) = \{ f(n) \mid \forall 常数 c > 0, \exists 常数 n_0 > 0 \ such \ that \ 0 \le cg(n) < f(n) \text{ for all } n \ge n_0 \}$

即：对任意常数$c > 0$，$cg(n)$对足够大的n要严格小于$f(n)$。

$\therefore f$和g必定不是同数量级，(f量级$>$ g量级)

- 例：$\frac{n^2}{2} = \omega(n)$ 但 $\frac{n^2}{2} \neq \omega(n^2)$

$$f(n) = \omega(g(n)) \Rightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$